

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

**Absolvování individuální odborné praxe**  
**Individual Professional Practice in the**  
**Company**

2018

Jakub Dlouhý

## Zadání bakalářské práce

Student:

**Jakub Dlouhý**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Absolvování individuální odborné praxe  
Individual Professional Practice in the Company

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: Vebvalley s.r.o.
2. Struktura závěrečné zprávy:
  - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
  - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
  - c) Zvolený postup řešení zadaných úkolů.
  - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
  - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
  - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

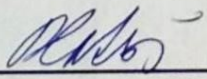
Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

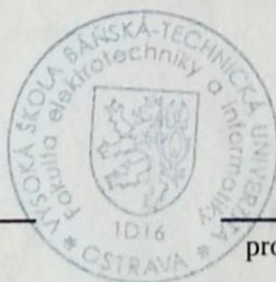
Vedoucí bakalářské práce: **doc. RNDr. Petr Šaloun, Ph.D.**

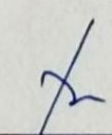
Konzultant bakalářské práce: Bc. Lukáš Vlček

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018

  
doc. Ing. Jan Platoš, Ph.D.  
vedoucí katedry



  
prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě, dne 7. dubna 2018

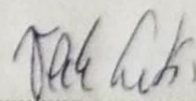
A handwritten signature in blue ink, consisting of stylized, cursive letters, likely representing the author's name.



### Prohlášení zástupce spolupracující právnické osoby

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě dne 26. 4. 2018



Bc. Lukáš Vlček,  
jednatel Webvalley s.r.o.

**webvalley**

Webvalley s.r.o.

Adresa:  
Pohraniční 504/27  
703 00 Ostrava

Kontakt:  
info@webvalley.cz  
www.webvalley.cz

IČO: 29 44 85 31  
DIČ: CZ 29 44 85 31

Chtěl bych poděkovat všem ve firmě Webvalley s.r.o. za vytvoření příjemné atmosféry pro vypracování této práce a také za jejich odbornou pomoc, když jsem narazil na jakékoli problémy během návrhu nebo implementace úkolů.

# Abstrakt

V této bakalářské práci popisuji průběh mé odborné praxe ve firmě Webvalley s.r.o. Je zde stručný popis firmy a jejího odborného zaměření. Práce zahrnuje i mé postavení ve firmě. Dále je zde popis jednotlivých dílčích úkolů, na kterých jsem pracoval a také postupy a technologie, které jsem během realizace úkolů použil. Většina mých úkolů se týkala tvorby a implementace analytických nástrojů pro prodejní weby. Nejmenší část potom tvořilo napojení prodejního webu na eCommerce řešení třetí strany, které klient používá pro svou správu prodeje a produktů. Během praxe jsem nabyl mnoho zkušeností, které jistě použiji ve svém profesním životě.

**Klíčová slova:** individuální odborná praxe, JavaScript, JS, MongoDB, Treant.js, analytické nástroje, UltraCart, Node.js, Express.js, eCommerce

# Abstract

In this bachelor thesis I describe the course of my professional practice in Webvalley s.r.o. There is a brief description of the company and its special focus. The work includes my position in the company. Here is a description of the individual tasks I worked on, as well as the procedures and technologies I used during the implementation of the tasks. Most of my tasks concerned the creation and implementation of analytics tools for sales sites. The smallest part then consisted of linking a sales site to third-party eCommerce solution that the client uses to manage sales and products. During my practice I have gained a lot of experience that I will certainly use in my professional life.

**Key words:** professional practice, JavaScript, JS, MongoDB, Treant.js, analytics tools, UltraCart, Node.js, Express.js, eCommerce

# Obsah

Seznam použitých zkratk a symbolů .....	8
Seznam obrázků .....	9
Seznam výpisů zdrojového kódu .....	10
1. Úvod .....	11
2. Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta .....	12
2.1 Popis odborného zaměření firmy .....	12
2.2 Pracovní zařazení studenta .....	12
3. Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti .....	13
3.1 Seznámení se s Node.js a s Express.js .....	13
3.2 Zpracování analytických dat a jejich zobrazení pomocí knihovny Treant .....	13
3.3 Analytický nástroj pro sběr, ukládání a zobrazení uživatelských dat .....	14
3.4 Implementace eCommerce platformy (Ultracart) na existující projekt .....	14
4. Zvolený postup řešení zadaných úkolů .....	15
4.1 Seznámení se s Node.js a s Express.js .....	15
4.2 Zpracování analytických dat a jejich zobrazení pomocí knihovny Treant .....	15
4.3 Analytický nástroj pro sběr, ukládání a zobrazení uživatelských dat .....	20
4.4 Implementace eCommerce platformy (Ultracart) na existující projekt .....	23
5. Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe .....	25
6. Znalosti či dovednosti scházející studentovi v průběhu odborné praxe .....	26
7. Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení .....	27
Literatura .....	28

# Seznam použitých zkratek a symbolů

- AJAX
  - Asynchronous JavaScript and XML
- API
  - Application Programming Interface
- CRON
  - Command run on
- CSS
  - Cascading Style Sheets
- CSV
  - Comma-separated values
- GDPR
  - General Data Protection Regulation
- OOP
  - Objektově orientované programování
- URL
  - Uniform Resource Locator



## Seznam obrázků

Obrázek 1: Node.js logo.....	12
Obrázek 2: Schéma pro cyklus událostí .....	15
Obrázek 3: Ukázka exportovaných událostí z Google Analytics .....	17
Obrázek 4: Úvodní stránka analytického nástroje.....	18
Obrázek 5: Vykreslený strom průchodů zákazníků .....	20
Obrázek 6: Stránka košíku .....	23

## Seznam výpisů zdrojového kódu

Ukázka 1: Vytváření permanentního URL.....	18
Ukázka 2: Funkce pro vytvoření úrovní stromu a stromu samotného.....	19
Ukázka 3: Modul pro kontrolu administrátorského účtu.....	21
Ukázka 4: CRON pro odesílání dat o zákaznících klientovi.....	23
Ukázka 5: Vytvoření košíku voláním Ultracart API.....	24

# 1. Úvod

V této bakalářské práci popisuji průběh mé odborné praxe ve firmě Webvalley s.r.o[1]. Důvod proč jsem se rozhodl jít cestou odborné praxe je, že stále více firem projevuje zájem o absolventy s praktickou znalostí oboru a dává jim přednost před absolventy s pouhou teoretickou znalostí. Protože tito studenti mají během své praxe příležitost získat zkušenosti s týmovou spoluprací, podáváním výkonu pod tlakem a vývojem softwaru pro reálné klienty. Většinou se jim i naskytne možnost zlepšit si své komunikační schopnosti. A celkově získají lepší přehled nad tím, jak vypadá běžný pracovní den vývojáře ve firmě, což může vést ke zlepšení některých vlastností studenta. Proto je pro firmy výhodnější zaměstnat absolventa s ověřenou praktickou znalostí.

Práce je převážně věnovaná tvorbě a implementaci analytických nástrojů pro prodejní weby. Obsahuje také implementaci eCommerce softwaru třetí strany pro zprovoznění eShopu.

V dalších kapitolách následuje popis firmy, ve které jsem bakalářskou praxi absolvoval a také popis jednotlivých úkolů a jejich řešení.

## 2. Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta

### 2.1 Popis odborného zaměření firmy

Firma Webvalley s.r.o. existuje již od roku 2012 se zaměřením na poskytování webových služeb. Jejím cílem je nabídnout klientům vše, co se webového vývoje týká od grafického návrhu přes vývoj samotné aplikace po analýzu návštěvníků jejich stránek. K webovým aplikacím také firma nabízí vývoj mobilních aplikací. Z důvodů automatizace a zlepšení efektivity práce firma vyvíjí i různé vlastní nástroje. Převážně se zde pracuje pro klienty, kteří nabízejí info produkty se zaměřením na zdravý životní styl nebo fitness. Firma má však klienty i v jiných odvětvích. Ze začátku probíhala propagace pouze na Americkém trhu, nyní však začíná přicházet do podvědomí i u nás v České republice. Sídlo firmy se nachází v Ostravě-Vítkovicích.

### 2.2 Pracovní zařazení studenta

I přes to, že firma své webové služby píše v PHP[2], tak jsem zastával roli JavaScript[3] vývojáře. To z důvodu, že pro analytické nástroje, které jsem měl za úkol vytvořit, byla vybrána technologie Node.js[4] pro kterou se píše kód v jazyce JavaScript.



Obrázek 1: Node.js logo

## 3. Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti

### 3.1 Seznámení se s Node.js a s Express.js

Mým prvním úkolem bylo seznámit se se softwarovým systémem Node.js. Kód pro tento systém se píše v JavaScriptu, s kterým jsem už v době nástupu na odbornou praxi měl nějaké zkušenosti. Avšak bylo zde ještě hodně k naučení, co se týče vnitřního fungování systému. Studování tohoto systému samotného nebyl časově moc náročný úkol. Bohužel bylo na firmě málo lidí, kteří měli takové zkušenosti, aby mi dokázali odpovědět na všechny mé otázky, které jsem měl ohledně některých principů. Proto byla moje práce převážně samostatná.

Ke studování Node.js šlo ruku v ruce i studium frameworku Express.js[5]. Časová náročnost pro tento úkol byla minimální díky flexibilní struktuře a intuitivnímu chování frameworku.

### 3.2 Zpracování analytických dat a jejich zobrazení pomocí knihovny Treant

Pro analytický tým jsem měl vymyslet způsob sbírání dat o chování zákazníků a následně vytvořit s pomocí Node.js nástroj, který by tato data dokázal zpracovat. Konkrétně bylo potřeba získat rozhodnutí zákazníků po tom, co učinili nákup produktu. Protože po nákupu jsou zákazníci přesměrováni na stránku, kde je jim nabízen další produkt za sníženou cenu. Zde se můžou rozhodnout, jestli tento produkt chtějí přidat do objednávky nebo ne. Po rozhodnutí jsou přesměrováni na další takovou stránku s další nabídkou. Toto přesměrování se může opakovat 5krát dokud zákazníci nedojdou na stránku s přehledem své objednávky. Můj úkol byl zachytit rozhodnutí zákazníků na těchto stránkách s nabídkami a následně pomocí již zmiňovaného nástroje vykreslit strom průchodů zákazníků těmito stránkami. Tento úkol byl hodně časově náročný, protože vymyšlení celého procesu sbírání i zpracování dat bylo složité.

### **3.3 Analytický nástroj pro sběr, ukládání a zobrazení uživatelských dat**

Pro tento úkol bylo zapotřebí vymyslet způsob sběru informací na prodejních stránkách jednoho z klientů Webvalley o nových zákaznících a následně pomocí webového rozhraní tato data vypisovat. Také bylo za potřebí ve formátu CSV každý den posílat nově získané zákazníky klientovi do e-mailové schránky. Tento úkol byl časově středně náročný.

### **3.4 Implementace eCommerce platformy (Ultracart) na existující projekt**

Mým posledním úkolem bylo napojení nového eShopu klienta na eCommerce platformu Ultracart[6]. Konkrétně se jednalo o napojení nákupního procesu. Co se týče složitosti, tak tento úkol nebyl tak náročný jako vytváření analytických nástrojů, ale byl časově hodně náročný, protože se muselo myslet na všechny možné scénáře, které se nám mohly s platformou Ultracart naskytnout.

## 4. Zvolený postup řešení zadaných úkolů

### 4.1 Seznámení se s Node.js a s Express.js

Node.js je běhové prostředí pro psaní vysoce škálovatelných webových aplikací. Krom webu se dá použít i pro další účely. Node.js používají například různé pre-procesory pro CSS a JavaScript nebo další užitečné nástroje, aby ulehčily práci vývojářům webových frontendů. Dokonce se již dají psát i desktopové aplikace za pomoci tohoto prostředí. Jako jazyk pro vývoj se zde používá JavaScript. Prostor je postavené na JavaScript engine V8 od firmy Google. Běh programu je omezen pouze na jedno vlákno a je řízen cyklem událostí.



Obrázek 2: Schéma pro cyklus událostí

Express.js je minimalistický framework pro webové aplikace postavené na Node.js. Je známý hlavně svou flexibilní strukturou, velkou knihovnou dostupných modulů a principem middlewaru. Díky middlewaru je možné prohnat požadavek uživatele skrze sérii funkcí, které ovlivňují samotný požadavek i odpověď serveru.

### 4.2 Zpracování analytických dat a jejich zobrazení pomocí knihovny Treant

Pro zjištění rozhodnutí uživatelů jsme se rozhodli využít Google Analytics[7] události. Platforma, kterou klient používá pro platby a administraci produktů se zároveň stará i o přesměrování zákazníka na další nabídku v procesu, kdy jsou zákazníkovi nabízeny další produkty, kterými si může obohatit svou objednávku. Během tohoto procesu máme možnost předávání vlastních parametrů mezi



prodejními stránkami. Tuto možnost jsem se rozhodl využít pro předávání dosavadních rozhodnutí zákazníka. Na každé prodejní stránce posílám všechny dosavadní rozhodnutí zákazníka pro případ, že by zákazník odešel před dosažením stránky se souhrnem objednávky. Jednotlivé zákazníky rozlišuji na základě ID jejich objednávky. Tímto je zajištěno, že budou vždy události unikátní pro každého zákazníka a zároveň jsem se tímto vyhnul posílání osobních údajů, což je s ohledem na přicházející GDPR velkou výhodou. Odesílání rozhodnutí se vždy provádí po rozhodnutí zákazníka o tom, jestli si chce produkt přidat do objednávky a před tím než je přesměrován na další stránku s nabídkou nebo na stránku se souhrnem. S každou odeslanou událostí se odesílají i dodatečné údaje o nákupním procesu, aby mohl analytik lépe vyfiltrovat události ve webovém rozhraní Google Analytics. Nejedná se však o žádné osobní údaje. Celý proces může vypadat takto:

1. Zákazník provede nákup produktu
2. Zákazník je po zaplacení přesměrován na dodatečnou nabídku dalšího produktu
3. Zákazník přijme/odmítne nabídku kliknutím na dané tlačítko.
  - a. Tlačítko pro přijetí nabídky nastavuje parametr rozhodnutí do URL na hodnotu „A“ (Accepted)
  - b. Tlačítko pro odmítnutí nabídky nastavuje parametr rozhodnutí do URL na hodnotu „D“ (Declined)
4. Proveďte se odeslání události do Google Analytics
  - a. Parametr rozhodnutí obsahuje „A“ nebo „D“
5. Zákazník je přesměrován na stránku s další nabídkou
6. Zákazník přijme/odmítne nabídku kliknutím na dané tlačítko.
  - a. Tlačítko pro přijetí nabídky má již v URL parametr rozhodnutí z předchozí stránky a k ní přidává „A“
  - b. Tlačítko pro odmítnutí nabídky má již v URL parametr rozhodnutí z předchozí stránky a k ní přidává „D“
7. Proveďte se odeslání události do Google Analytics
  - a. Parametr rozhodnutí obsahuje „AA“ nebo „DD“ nebo „AD“ nebo „DA“ podle toho jak se zákazník rozhodl v předchozím kroku.
8. Zákazník je přesměrován na souhrn objednávky

Díky tomuto mechanismu teď může analytik exportovat z Google Analytics výpis událostí ve formátu CSV, se kterým se dá dále pracovat.

	A	B	C	D	E	F
1	# -----					
2	# All Web Site Data					
3	# Top Events					
4	# 20170601-20170621					
5	# -----					
6						
7	Event Label	Event Action	Total Ever	Unique Ev	Event Val	Avg. Value
8	HG8QGVKE6F-AD	boostmonkitpd-29072	12	1	0	0.00
9	HG8QG48EHK-D	boostmonkitpd-29072	8	1	0	0.00
10	WCNLGTRE9B-D	boostmonkitpd-29072	8	1	0	0.00
11	HG8QGHHE9F-DA	boostmonkitpd-29072	6	1	0	0.00
12	5DFFAK2ZB6-ADA	boostmonkitpd-29072	4	1	0	0.00
13	6K5IQG6K6T-A	boostmonkitpd-29072	4	1	0	0.00
14	8ATM2V3DTG-D	boostmonkitpd-29072	4	1	0	0.00
15	A8N2ZNHCDF-A	boostmonkitpd-29072	4	1	0	0.00
16	HG8QGVKE6F-A	boostmonkitpd-29072	4	1	0	0.00
17	SWLILG3FZB-DD	boostmonkitpd-29072	4	1	0	0.00
18	SWLILG7PNT-A	boostmonkitpd-29072	4	1	0	0.00
19	3B5QG7LE6K-D	boostmonkitpd-29072	3	1	0	0.00
20	5DFFAK2GRM-DDDD	boostmonkitpd-29072	3	1	0	0.00
21	A64QZN68SQ-A	boostmonkitpd-29072	3	1	0	0.00
22	A8N2ZKNX4N-D	boostmonkitpd-29072	3	1	0	0.00
23	A8N2ZNMXP-D-A	boostmonkitpd-29072	3	1	0	0.00
24	AD7KZNP63Z-DD	boostmonkitpd-29072	3	1	0	0.00
25	AD7KZNS53T-D	boostmonkitpd-29072	3	1	0	0.00
26	D2LQG37EYL-D	boostmonkitpd-29072	3	1	0	0.00
27	HG8QG3DE2G-D	boostmonkitpd-29072	3	1	0	0.00
28	HG8QGPLEBG-DD	boostmonkitpd-29072	3	1	0	0.00
29	HG8QGXPE83-DADA	boostmonkitpd-29072	3	1	0	0.00

**Obrázek 3: Ukázka exportovaných událostí z Google Analytics**

Po dokončení implementace sběru dat jsem začal pracovat na nástroji, který exportované CSV dokázal zpracovat a pomocí dat vytvořit strom, který by znázorňoval průchod zákazníků prodejními stránkami. Tento nástroj jsem realizoval pomocí frameworku Express.js, který běží v běhovém prostředí Node.js. První krok po nastavení Express aplikace bylo vytvoření úvodní stránky. Vzhledem k tomu, že se jedná o interní nástroj, který bude používat pouze náš analytický tým, tak jsou všechny stránky nástroje velice jednoduché v oblasti designu. Na úvodní stránce je jen možnost nahrání souboru a odesílací tlačítko.

# Upsell Statistics

Welcome to Upsell Statistics

Vybrat soubor

Soubor nevybrán

Odeslat

Obrázek 4: Úvodní stránka analytického nástroje

Po odeslání formuláře se CSV soubor uloží na server a začne se zpracovávat. Aplikace nepoužívá databázi, ale i tak se po každém zpracování souboru vytvoří permanentní odkaz, který vždy povede na daný výsledek analýzy. Toho jsem docílil tím, že po každém nahrání souboru vytvořím objekt, který obsahuje název souboru a datum nahrání. Tento objekt převedu na textový řetězec a ten zakóduji pomocí Base64. Zakódovaný řetězec po té použiji v URL na daný výsledek.

```
1. uploadCSV: function(req, res) {
2.   console.log(req.file);
3.
4.   if(req.file) {
5.     var info = base64url.encode(JSON.stringify({ file: req.file.filename, date:
6.       (new Date()).toUTCString() }));
7.     res.redirect(routes.index.processCSV.replace(':file', info));
8.   }
9.   else
10.    res.redirect(routes.index.homepage);
11. },
```

Ukázka 1: Vytváření permanentního URL

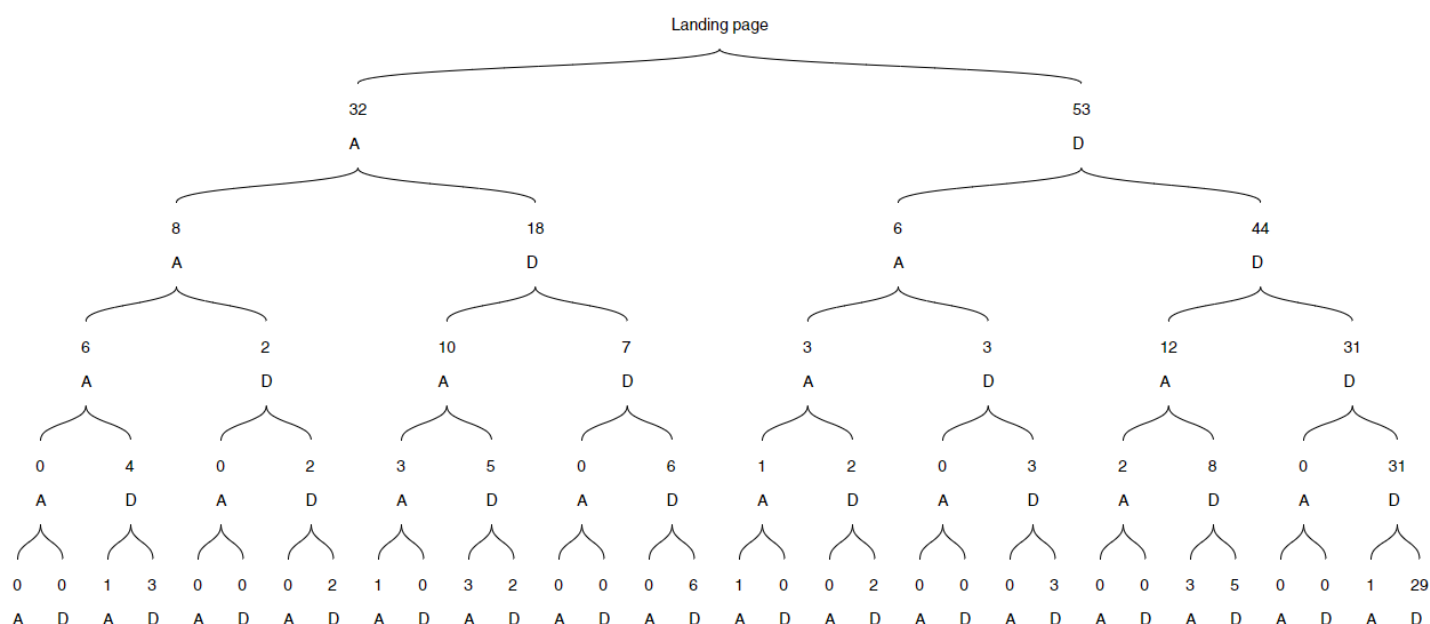
Následné zpracování souborů probíhá tak, že funkce rozkóduje objekt z URL a pomocí něj získá jméno souboru na serveru. Tento soubor otevřu a procházím řádek po řádku a zpracovávám jednu událost za druhou. Z každé události si uložím sledovací řetězec, který se hodí analytikovi pro rozpoznání toho, na jakých stránkách zákazníci byli. Nejdůležitější je však ukládání rozhodnutí zákazníků. Proto vytvářím objekt, kde jako klíč slouží ID objednávky zákazníka a hodnotu tvoří parametr rozhodnutí. Při průchodu událostí kontroluji, jestli již zákazníkovo ID v objektu existuje. Pokud neexistuje, tak jej i s parametrem rozhodnutí přidám. Pokud již dané ID v objektu existuje, zkontroluji, jestli je uložený parametr rozhodnutí delší než parametr ze současné události a uložím ten

nejdelší. Tímto způsobem se snažím získat od každého uživatele nejvíce rozhodnutí, aby výsledný strom obsahoval všechny průchody. Poslední parametr, který během průchodu událostí ukládám, je počet stránek s dodatečnými nabídkami. Tento parametr slouží k tomu, aby se správně vygenerovaly úrovně stromu. Po dokončení průchodu událostmi ze získaných dat vytvořím kostru stromu, kterou po té naplním počty průchodů.

```
1. function generateLevels(customers, max) {
2.     var levels = [];
3.
4.     for(var i = 0; i < max; i++)
5.         levels[i] = new Set();
6.
7.     Object.keys(customers).forEach(function (key) {
8.         let tempCustomer = customers[key];
9.
10.        for(var i = 0; i < tempCustomer.length; i++)
11.            levels[i].add(tempCustomer[i]);
12.
13.    });
14.
15.    for(var i = 0; i < max; i++)
16.        levels[i] = [...levels[i]].sort();
17.
18.    return levels
19. }
20.
21. function generateTree(levels, index) {
22.     var current = levels[index];
23.     var result = [];
24.
25.     if(index < levels.length - 1)
26.         for(var i = 0; i < current.length; i++)
27.             result.push({
28.                 text: {
29.                     name: 0,
30.                     title: current[i]
31.                 },
32.                 children: generateTree(levels, index + 1)
33.             })
34.     else
35.         for(var i = 0; i < current.length; i++)
36.             result.push({
37.                 text: {
38.                     name: 0,
39.                     title: current[i]
40.                 }
41.             })
42.
43.     return result
44. }
```

## Ukázka 2: Funkce pro vytvoření úrovní stromu a stromu samotného

Pro analytika je strom generován pomocí knihovny Treant.js[8], který z daného objektu dokáže vykreslit přehledný strom, kde jdou vidět všechny průchody.



V současné době je aplikace denně používána pro zpracování dat pro prezentaci klientovi, čímž analytikům šetří spoustu času. V brzké budoucnosti se plánuje její rozšíření o další analytické nástroje, které by měly mít za úkol více zefektivnit práci analytického týmu.

### 4.3 Analytický nástroj pro sběr, ukládání a zobrazení uživatelských dat

Pro sběr dat uživatelů jsem se v tomto úkolu rozhodl, že půjdu cestou vlastního API, kam se budou data odesílat. Web klienta má dvoufázový proces objednávky. První formulář se nachází přímo na stránce s informacemi o produktu, což je první stránka, na kterou se zákazník dostane. Tento formulář obsahuje všechny informace potřebné k odeslání balíčku. Po odeslání tohoto formuláře se zákazník dostane na stránku s druhým formulářem, kde je požádán o vyplnění platebních údajů.

Sběr dat uživatelů provádím před odesláním formuláře na server. Používám k tomu nativní posluchač událostí, který vyše signál před tím než se začne formulář odesílat, čímž mi dovolí na tento signál zareagovat a získat všechny data, které zákazník vyplnil a pomocí AJAXu je odeslat do aplikace, která tyto data uloží do databáze.

Pro aplikaci, která běží na serveru a sbírá uživatelská data, jsem si vybral framework Express.js. Pro ukládání dat jsem vytvořil MongoDB[9] instanci. Tuto dvojici technologií jsem vybral z důvodu jednoduché komunikace mezi aplikací a databází pomocí modulu třetí strany. Vzhledem k tomu, že skrze aplikaci jsou přístupná osobní data uživatelů, tak bylo potřeba vytvořit přihlašovací

systém, aby nebyly tyto data volně dostupné skrze webový prohlížeč. Pro tento účel jsem použil knihovnu „Passport“ [10], která jen stačila nakonfigurovat a byla použitelná pro přihlašování do systému pomocí přihlašovacího jména a hesla. Abych se ujistil, že bude vždy k dispozici administrátorský účet, tak jsem vytvořil skript, který kontroluje existenci „admin“ účtu a v případě jeho absence tento účet vytvoří.

```
1. 'use strict'
2. var User = require('../app/models/userModel');
3. var bcrypt = require('bcrypt-nodejs');
4.
5. // Generates hash using bcrypt
6. var createHash = function(password){
7.     return bcrypt.hashSync(password, bcrypt.genSaltSync(10), null);
8. }
9.
10. module.exports = function() {
11.     User.findOne({ 'username' : 'admin' }, function(err, user) {
12.         // In case of any error, return using the done method
13.         if (err){
14.             console.log('Error in SignUp of Administrator: '+err);
15.         }
16.
17.         if(!user) {
18.             var newUser = new User();
19.
20.             newUser.username = 'admin';
21.             newUser.password = createHash('test');
22.             newUser.email = 'test@test.cz';
23.
24.             // save the user
25.             newUser.save(function(err) {
26.                 if (err){
27.                     console.log('Error in Saving administrator: '+err);
28.                     throw err;
29.                 }
30.                 console.log('Administrator Registration succesful');
31.             });
32.         }
33.     });
34. }
```

### Ukázka 3: Modul pro kontrolu administrátorského účtu

Vytvoření API endpointu pro sběr dat doprovázela jedna menší nepříjemnost. Tou bylo výchozí chování Express aplikace na volání z jiné URL, než na které aplikace běží. Tato nepříjemnost byla naštěstí snadno opravitelná pomocí modulu „cors“ [11], který zjednodušuje nastavení povolených zdrojů. Samotný endpoint nebylo těžké vytvořit, protože se jedná pouze o funkci, která uloží data, která dostane.

Složitější částí tohoto úkolu byl export zákazníků do CSV a odesílání nových zákazníků na email klienta. Díky tomu, že Node.js je stále živý organismus, tak se dají CRON operace přepsat přímo v samotné aplikaci a nemusí se nastavovat na serveru jako je tomu například u PHP. K tomuto účelu

vytvořil náš správce serverů emailový server, přes který jsme odesílali klientovi emaily s daty. Klient chtěl dostávat tyto data každý den ve 12 PM.

```
1. 'use strict'
2. const nodemailer = require('nodemailer');
3. var schedule = require('node-schedule');
4. var path = require('path');
5. var fs = require('fs');
6. var Data = require('../..app/models/dataModel');
7.
8. let transporter = nodemailer.createTransport({
9.   host: 'smtp.server.com',
10.  auth: {
11.    user: '#####',
12.    pass: '#####'
13.  },
14.  tls: {
15.    // do not fail on invalid certs
16.    rejectUnauthorized: false
17.  }
18. });
19.
20. // verify connection configuration
21. transporter.verify(function(error, success) {
22.   if (error) {
23.     console.log(error);
24.   } else {
25.     console.log('Server is ready to take our messages');
26.   }
27. });
28.
29. let mailOptions = {
30.   from: '"Webvalley" <info@webvalley.cz>', // sender address
31.   to: 'info@ruschbiotech.com', // list of receivers
32.   subject: 'Data from TelXtend', // Subject line
33.   text: 'Collected data', // plain text body
34.   html: '<b>Collected data</b>', // html body
35.   attachments: [
36.     { // filename and content type is derived from path
37.       path: path.join(__dirname, '../..public/current.csv')
38.     },
39.   ]
40. };
41.
42. module.exports = function() {
43.   var job = schedule.scheduleJob('0 12 * * *', function() {
44.     var d = new Date();
45.     d.setDate(d.getDate() - 1);
46.
47.     Data.findAndStreamCsv({ created: { $gte: d } })
48.       .pipe(fs.createWriteStream('public/current.csv'))
49.       .on('finish', function() {
50.         transporter.sendMail(mailOptions, (error, info) => {
51.           if (error) {
52.             return console.log(error);
53.           }
54.
55.           console.log('Message %s sent: %s', info.messageId, info.respon
56.             });
57.         });
58.       });
59.   });
60. }
```



```

57.         })
58.     );
59. });
60. }

```

#### Ukázka 4: CRON pro odesílání dat o zákaznících klientovi

V současné době je činnost této aplikace pozastavena z důvodu přerušení spolupráce s klientem. Do budoucna se s rozšiřováním funkcionalit nepočítá.

## 4.4 Implementace eCommerce platformy (Ultracart) na existující projekt

Tento úkol byl součástí přepisování současného eShopu klienta, kdy se všechny stránky psaly od začátku. Napojení na API jsem psal v JavaScriptu. Celý nákupní proces začíná tím, že si uživatel zvolí produkt na výpisu produktů, což ho přeměruje na detail produktu. Zde si vybere, kolik lahviček by chtěl přidat do košíku. ID produktu s počtem kusů se uloží do cookies, kde zůstává pro pozdější použití. V tomto kroku se zatím nevolá žádné API, protože to není nutné. Když se uloží produkt do cookies, tak je uživatel přesměrován na stránku košíku, kde může dokončit objednávku. Zde se už API volá s tím, že pouze zadám, které produkty si chce uživatel koupit a Ultracart mi vrátí celý objekt košíku pomocí kterého vykreslím produkty v košíku, možnosti dopravy, platební metody a finální ceny.

Obrázek 6: Stránka košíku

Položky jdou i na stránce košíku odebrat z objednávky. Při vytváření skriptů pro tuto stránku jsem měl za cíl mít všechny dynamické části košíku získané pomocí API, aby mohl klient co nejvíce věci ovlivnit ve webovém rozhraní Ultracartu. Problém nastal v případě, kdy jsem využíval API pro získání států dané země, kam si může zákazník nechat zaslat svůj balíček. Vše spočívalo v tom, že API nepodporovalo Velkou Británii, takže jsem musel manuálně vytvořit soubor, který nahradil odpověď z API.

```
1. function createCart() {
2.     var products = readCookie("products");
3.     if(products != null && products != "") {
4.         var xhttp = new XMLHttpRequest();
5.         xhttp.onreadystatechange = function() {
6.             if (xhttp.readyState == 4 && xhttp.status == 200) {
7.                 var data = JSON.parse(xhttp.responseText);
8.                 createCookie("UltraCartShoppingCartID", data.cartId);
9.                 createCookie("UltraCartMerchantID ", "vikas");
10.
11.                 updateCart(true);
12.             }
13.         };
14.
15.         xhttp.open("GET", "https://vikas.ultracartstore.com/rest/cart", true);
16.         xhttp.setRequestHeader("X-UC-Merchant-Id", "vikas");
17.         xhttp.setRequestHeader("cache-control", "no-cache");
18.         xhttp.send();
19.     }
20. }
```

#### Ukázka 5: Vytvoření košíku voláním Ultracart API

Skript pro práci s Ultracart API je nyní využíván na více speciálních stránkách tohoto klienta, kde byl lehce upraven, aby vyhovoval požadavkům speciálních stránek. Do budoucna se nadále počítá s jeho využíváním a případným rozšířením.

## **5. Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe**

Během výkonu mé praxe jsem nejvíce využil předměty, kde se pracovalo s programovacím jazykem JavaScript, protože jsem v tomto jazyce psal všechny skripty a aplikace popsané v této práci. Konkrétně se jedná o předměty TAMZ I a VIA. V těchto předmětech jsem se naučil specifické vlastnosti tohoto jazyka a jeho odlišnost od jazyků, s kterými jsem doposud přišel do styku. Ostatní programovací předměty mě naučili správným zásadám psaní čitelného kódu a pochopení principu OOP, což bylo během vytváření aplikací také velkou výhodou.

Využití jsem našel i pro znalosti a zkušenosti ze všech databázových předmětů, díky kterým jsem byl schopen navrhnout a následně i vytvořit databázi pro sběr informací.

Předměty, ve kterých jsem se naučil základy webu a webových serverů, byly taky velkým přínosem, protože díky těmto znalostem jsem měl základní přehled o odborném zaměření firmy.

V poslední řadě jsem využil znalost z programovacích předmětů, kde jsem se naučil principům API a měl jsem možnost získat i praktické zkušenosti s voláním vzdáleného serveru a zpracováváním odpovědi, kterou mi server poskytl. Tyto znalosti a zkušenosti jsem používal napříč celou mou prací.

## **6. Znalosti či dovednosti scházející studentovi v průběhu odborné praxe**

Co se týče technických znalostí a dovedností, tak mi scházelo povědomí o Node.js a frameworku Express.js, které jsem hojně ve své práci využíval. Také mi scházela znalost různých měřicích technik pro prodejní weby. Do těchto znalostí počítám nastavení a implementaci měřicích pixelů a posílání událostí do analytických nástrojů třetích stran. Tyto techniky jsou v oboru tvorby webových aplikací hodně využívány.

## 7. Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Díky této odborné praxi jsem měl možnost vyzkoušet si reálné pracovní nasazení a zjistit jak vypadá běžný den vývojáře ve firmě. Nabyl jsem mnoho zkušeností s jazykem JavaScript a dalšími technologiemi, které tento jazyk využívají. Což zlepšilo i mé celkové schopnosti jako programátora. Naučil jsem se práci s verzovacím systémem Git[12].

Co se týče mimo technických zkušeností, tak jsem se naučil práci v týmu a zlepšil jsem své komunikační schopnosti. Dále jsem si uvědomil, že v reálném životě programátora někdy nestačí pouze umět psát čistý a funkční kód a rozumět technologiím, s kterými programátor pracuje. Ale taky je důležité, aby měl programátor ponětí o tom, pro jaký účel bude daný software sloužit. Proto jsem si z mé praxe vzal i mnoho znalostí o tom, jak funguje byznys model našich klientů. Tímto porozuměl, která data jsou pro ně důležitá a na základě toho jsem mohl postavit analytické nástroje, které mají reálnou hodnotu.

Mou odbornou praxi hodnotím velice kladně, protože jsem měl možnost řešit reálné problémy a dostal jsem možnost vytvořit nástroje, které se používají pro reálné klienty a jejich potřeby. Rozhodně se nebojím říct, že pro můj osobní i kariérní růst byla tato praxe velkým přínosem. Díky tomu ve firmě i po odpracování dnů potřebných pro tuto praxi zůstávám a budu nadále rozšiřovat nástroj pro analýzu uživatelských dat a budu zařazen i mezi jiné projekty firmy.

# Literatura

- [1] Webvalley s.r.o. [online]. [cit. 2018-04-05]. Dostupné z: <https://webvalley.cz/>
- [2] PHP: Hypertext Preprocessor [online]. [cit. 2018-04-05]. Dostupné z: <http://php.net/>
- [3] JavaScript [online]. [cit. 2018-04-05]. Dostupné z: <https://www.javascript.com/>
- [4] Node.js: JavaScript runtime [online]. [cit. 2018-04-05]. Dostupné z: <https://nodejs.org/en/>
- [5] Express.js: Fast, unopinionated, minimalist web framework for Node.js [online]. [cit. 2018-04-05]. Dostupné z: <https://expressjs.com/>
- [6] Ultracart [online]. [cit. 2018-04-05]. Dostupné z: <https://www.ultracart.com/>
- [7] Google Analytics [online]. [cit. 2018-04-05]. Dostupné z: <https://www.google.com/analytics/>
- [8] Treant.js: JavaScript library for visualization of tree diagrams [online]. [cit. 2018-04-05]. Dostupné z: <http://fperucic.github.io/treant-js/>
- [9] MongoDB [online]. [cit. 2018-04-05]. Dostupné z: <https://www.mongodb.com/>
- [10] PassportJS: Simple, unobtrusive authentication for Node.js [online]. [cit. 2018-04-05]. Dostupné z: <http://www.passportjs.org/>
- [11] CORS: a node.js package for providing a Connect/Express middleware that can be used to enable CORS with various options. [online]. [cit. 2018-04-05]. Dostupné z: <https://www.npmjs.com/package/cors>
- [12] Git [online]. [cit. 2018-04-05]. Dostupné z: <https://git-scm.com/>